

Testing Mutual Duality of Planar Graphs

Patrizio Angelini¹, Thomas Bläsius², and Ignaz Rutter²

¹*Università Roma Tre, Italy*

angelini@dia.uniroma3.it

²*Faculty of Informatics, Karlsruhe Institute of Technology (KIT), Germany*

{thomas.blaesius,rutter}@kit.edu

Abstract

We introduce and study the problem **MUTUAL PLANAR DUALITY**, which asks for two planar graphs G_1 and G_2 whether G_1 can be embedded such that its dual is isomorphic to G_2 . Our algorithmic main result is an NP-completeness proof for the general case and a linear-time algorithm for biconnected graphs.

To shed light onto the combinatorial structure of the duals of a planar graph, we consider the *common dual relation* \sim , where $G_1 \sim G_2$ if and only if they have a common dual. While \sim is generally not transitive, we show that the restriction to biconnected graphs is an equivalence relation. In this case, being dual to each other carries over to the equivalence classes, i.e., two graphs are dual to each other if and only if any two elements of their respective equivalence classes are dual to each other. To achieve the efficient testing algorithm for **MUTUAL PLANAR DUALITY** on biconnected graphs, we devise a succinct representation of the equivalence class of a biconnected planar graph. It is similar to SPQR-trees and represents exactly the graphs that are contained in the equivalence class. The testing algorithm then works by testing in linear time whether two such representations are isomorphic.

We note that a special case of **MUTUAL PLANAR DUALITY** is testing whether a graph G is self-dual. Our algorithm handles the case where G is biconnected and our NP-hardness proof extends to testing self-duality of general planar graphs and also to testing map self-duality, where a graph G is map self-dual if it admits a planar embedding \mathcal{G} such that G^* is isomorphic to G , and additionally the embedding induced by \mathcal{G} on G^* is \mathcal{G} .

1 Introduction

Let G be a planar graph with planar embedding \mathcal{G} and let F be the set of faces of \mathcal{G} . The *dual* of G with respect to \mathcal{G} is the graph $G^* = (F, E^*)$, where $E^* = \{e^* \mid e \in E\}$ and e^* denotes the edge connecting the two faces incident to e in \mathcal{G} . Thus G^* models the adjacencies of faces of G in the embedding \mathcal{G} . Note that the circular order of edges around faces in \mathcal{G} naturally induces a planar embedding \mathcal{G}^* on G^* and that the dual of G^* with respect to \mathcal{G}^* is G .

We consider the following problem, that we call **MUTUAL PLANAR DUALITY**. Given two planar graphs G_1 and G_2 , is it possible to find an embedding \mathcal{G}_1 of G_1 such that the dual G_1^* of G_1 with respect to \mathcal{G}_1 is isomorphic to G_2 ? If G_1 is triconnected it has a fixed planar embedding [12] and thus the problem **MUTUAL PLANAR DUALITY** reduces to testing graph isomorphism for planar graphs, which can be done in linear time due to Hopcroft and Wong [6]. Observe that biconnectivity and triconnectivity of a planar graph is invariant under dualization [11]. For non-triconnected planar graphs **MUTUAL PLANAR DUALITY** is more complicated since changing the embedding of G_1 influences the structure of its dual graph. In fact, we show that **MUTUAL PLANAR DUALITY** is NP-complete in general.

On the other hand, for biconnected planar graphs we provide a linear-time algorithm solving **MUTUAL PLANAR DUALITY** that is based on the definition of a new data structure that we call *dual SPQR-tree* in analogy to the SPQR-tree [4, 5]. As SPQR-trees allow to succinctly represent and efficiently handle all the planar embeddings of a biconnected planar graph, the dual SPQR-trees, together with a newly-defined set of operations, allow to succinctly represent and efficiently handle all the dual graphs of a biconnected planar graph. This data structure has an interesting implication on the structure of the dual graphs of a biconnected planar graph. Namely, consider the *common dual relation* \sim , where $G_1 \sim G_2$ if and only if they have a common dual graph. We show that, \sim is not transitive on the set of connected planar graphs. However, it follows from the dual SPQR-tree that \sim is an equivalence relation on the set of biconnected planar graphs. In particular, the graphs represented by a dual SPQR-tree form an equivalence class. Thus, testing **MUTUAL PLANAR DUALITY** reduces to testing whether two dual SPQR-trees represent the same equivalence class.

We believe that this new data structure can be successfully used to efficiently solve other related problems. In fact, in many applications it is desirable to find an embedding of a given planar graph that optimizes certain criteria, and such criteria can often be naturally described in terms of the dual graph with respect to the chosen embedding. For example, Bienstock and Monma [3], and Angelini et al. [1] seek for an embedding of a given planar graph minimizing the largest distance of internal faces to the external face. In terms of the dual graph this corresponds to minimizing the largest distance of a vertex to all other vertices. Hence, for problems of this kind it might be useful to work directly with a representation of all dual graphs, that can be given by the dual SPQR-trees, instead of taking the detour over a representation of all planar embeddings, that is given by the original SPQR-trees.

We finally remark that the **MUTUAL PLANAR DUALITY** problem we introduce in this paper is a generalization of the self-duality of planar graphs [9]. A graph G is *graph self-dual* if it admits an embedding such that its dual G^* is isomorphic to G . We call the corresponding decision problem **GRAPH SELF-DUALITY**. A stronger form of self-duality can be defined as follows. A graph G is *map self-dual* [10] if and only if G has an embedding \mathcal{G} such that there exists an isomorphism from G to its dual graph G^* that preserves the embedding \mathcal{G} . The corresponding decision problem is called **MAP SELF-DUALITY**. Note that, since triconnected planar graphs have a unique planar embedding, **GRAPH SELF-DUALITY** and **MAP SELF-DUALITY** are equivalent for this class of graphs. Servatius and Servatius [10] show the existence of biconnected planar graphs that are graph self-dual but not map self-dual. Servatius and Christopher [9] show how to construct self-dual graphs from given planar graphs. Archdeacon and Richter [2] give a set of constructions for triconnected self-dual graphs and show that every such graph can be constructed in this way. To the best of our knowledge the computational complexity of testing **MAP** or **GRAPH SELF-DUALITY** is open. Since **GRAPH SELF-DUALITY** is a special case of **MUTUAL PLANAR DUALITY** (simply set $G_1 = G_2 = G$), our algo-

rithm solving MUTUAL PLANAR DUALITY in linear time can be used to solve GRAPH SELF-DUALITY in linear time when G is biconnected. Moreover, our construction showing NP-hardness of MUTUAL PLANAR DUALITY for general instances extends to MAP and GRAPH SELF-DUALITY.

Outline. In Section 2 we show that MUTUAL PLANAR DUALITY is NP-complete, even if both input graphs are required to be simple. With a similar construction we can show that MAP SELF-DUALITY and GRAPH SELF-DUALITY are NP-complete in general. To solve MUTUAL PLANAR DUALITY efficiently for biconnected graphs, we first describe decomposition trees as a generalization of SPQR-trees in Section 3. In Section 4 we describe the dual SPQR-tree and show that it succinctly represents all dual graphs of a biconnected planar graph. We consider the common dual relation in Section 5 and give a counter example showing that \sim is not transitive on the set of connected planar graphs. On the other hand, we show that it follows from the dual SPQR-tree that \sim is an equivalence relation on the set of biconnected planar graphs. This implies that solving MUTUAL PLANAR DUALITY is equivalent to testing whether two dual SPQR-trees represent the same equivalence class. In Section 6 we show that this can be further reduced to testing graph isomorphism of two planar graphs, which leads to a linear-time algorithm for MUTUAL PLANAR DUALITY, including GRAPH SELF-DUALITY as a special case.

2 Complexity

In this section we first show that MUTUAL PLANAR DUALITY is NP-complete by a reduction from 3-PARTITION. Then we show that the resulting instances of MUTUAL PLANAR DUALITY can be further reduced to equivalent instances of MAP and GRAPH SELF-DUALITY. An instance (A, B) of 3-PARTITION consists of a positive integer B and a set $A = \{a_1, \dots, a_{3m}\}$ of $3m$ integers with $B/4 < a_i < B/2$ for $i = 1, \dots, 3m$. The question is whether A admits a partition \mathcal{A} into a set of triplets such that for each triplet $X \in \mathcal{A}$ we have $\sum_{x \in X} x = B$. The problem 3-PARTITION is strongly NP-hard [7], i.e., it remains NP-hard even if B is bounded by a polynomial in m .

Theorem 1. *MUTUAL PLANAR DUALITY is NP-complete, even if both graphs are simple.*

Proof. Clearly, MUTUAL PLANAR DUALITY is in NP as we can guess an embedding for graph G_1 and then check in polynomial time whether the corresponding dual is isomorphic to G_2 .

To show NP-hardness we give a reduction from 3-PARTITION. We first give a construction containing loops, afterwards we show how to get rid of them. Let (A, B) be an instance of 3-PARTITION with $|A| = 3m$. The graph G_1 contains a wheel of size m , i.e., a cycle v_1, \dots, v_m together with a center u connected to each v_i . Additionally, for each element $a_i \in A$ we create a star T_i with $a_i - 1$ leaves and connect its center to u ; see Figure 1(a). The graph G_2 is a wheel of size m along with B loops at every vertex except for the center; see Figure 1(b). We now claim that G_1 and G_2 form a YES-instance of MUTUAL PLANAR DUALITY if and only if (A, B) is a YES-instance of 3-PARTITION.

Suppose that there exists a partition \mathcal{A} of A . The embedding of the wheel in G_1 is fixed and it has exactly m faces incident to the wheels center u . The remaining degree of freedom is to decide the embedding of the trees T_i into these m faces. For each triplet $X = \{a_i, a_j, a_k\} \in \mathcal{A}$ we pick a distinct such face and embed T_i, T_j and T_k into it. Call the resulting embedding \mathcal{G}_1 and consider the dual G_1^* with respect to \mathcal{G}_1 . The wheel of G_1 determines a wheel of size m in G_1^* . Consider a tree T_i that is embedded in a face f . Since T_i contains a_i bridges, which are all embedded in f , the corresponding vertex of G_1^* has a_i loops; see Figure 1(c). Due to the construction each face contains exactly three trees with a total of B bridges. Thus G_1^* is isomorphic to G_2 .

Conversely, assume that we have an embedding \mathcal{G}_1 such that the dual G_1^* of G_1 with respect to \mathcal{G}_1 is isomorphic to G_2 . Again, the wheel in G_1 forms m faces incident to u , and since G_1^* is isomorphic to G_2 ,

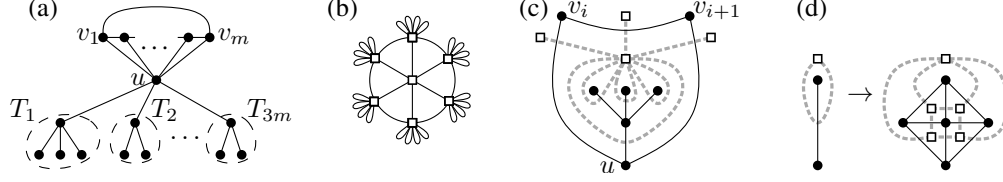


Figure 1: The graphs G_1 (a) and G_2 (b) of the reduction from 3-PARTITION. (c) Embedding a tree T_i inside a face f creates a_i loops at the corresponding dual vertex. (d) Bridges and corresponding loops can be replaced by small graphs.

the trees must be embedded such that each face contains exactly B bridges. Since embedding T_i inside a face f puts a_i bridges into f and we have $B/4 < a_i < B/2$ each face must contain exactly three trees. Thus the set of triplets determined by trees that are embedded in the same faces form a solution to 3-PARTITION.

Clearly, the transformation can be computed in polynomial time, and thus MUTUAL PLANAR DUALITY is NP-hard. Moreover, the graph G_2 can be made simple (G_1 is already simple) by replacing each bridge in G_1 and each loop in G_2 with a 4-wheel as depicted in Figure 1(d). The resulting graphs G'_1 and G'_2 are obviously dual to each other if and only if G_1 and G_2 are dual to each other. Moreover, G'_1 and G'_2 are simple, which concludes the proof. \square

In the following we show how the above construction can be used to show NP-completeness for MAP and GRAPH SELF-DUALITY. To this end, we use the *adhesion* operation introduced by Servatius and Christopher [9]. Let v be a vertex of G incident to a face f . Then the adhesion of G and its dual G^* (with respect to v and f) is obtained by identifying v in G and f^* in G^* with each other. Servatius and Christopher [9] show that the adhesion of a plane graph and its dual is graph self-dual. Moreover, they implicitly show that this adhesion is even map self-dual, although they do not mention it explicitly. To show the following theorem we essentially transform the instance of MUTUAL PLANAR DUALITY consisting of the two graphs G_1 and G_2 described in the proof of Theorem 1 into an equivalent instance of MAP and GRAPH SELF-DUALITY by forming the adhesion of G_1 and G_2 .

Theorem 2. GRAPH SELF-DUALITY and MAP SELF-DUALITY are NP-complete.

Proof. Clearly, GRAPH SELF-DUALITY (MAP SELF-DUALITY) is in NP as we can guess an embedding for G together with a bijection between the vertices of G and the vertices of G^* and then check in polynomial time whether this bijection is an isomorphism (that preserves the embedding).

Let G_1 and G_2 form an instance of MUTUAL PLANAR DUALITY obtained from an instance of 3-PARTITION as described in the proof of Theorem 1. Let G be the graph obtained from G_1 and G_2 by identifying a vertex that is not the center of the wheel in G_2 with the vertex u in G_1 . In the following we consider G as an instance of GRAPH SELF-DUALITY and MAP SELF-DUALITY. We claim the following.

Claim 1 If G is a YES-instance of MAP SELF-DUALITY, it is a YES-instance of GRAPH SELF-DUALITY.

Claim 2 If G_1 and G_2 form a YES-instance of MUTUAL PLANAR DUALITY, then G is a YES-instance of MAP SELF-DUALITY.

Claim 3 If G is a YES-instance of GRAPH SELF-DUALITY, then G_1 and G_2 form a YES-instance of MUTUAL PLANAR DUALITY.

The three claims together show that the instance G_1 and G_2 of MUTUAL PLANAR DUALITY, the instance G of GRAPH SELF-DUALITY and the instance G of MAP SELF-DUALITY are equivalent.

Claim 1 is clear since being map self-dual is a stricter requirement than being graph self-dual. For Claim 2 assume that G_1 and G_2 form a YES-instance of MUTUAL PLANAR DUALITY, that is G_1 and G_2 admit embeddings such that they are dual to each other. As the vertex u is incident to all faces in G_1 except for the face forming the center of the wheel in G_1^* , it is in particular incident to the face dual to the vertex

in G_2 chosen for the adhesion. Thus it follows from the results by Servatius and Christopher [9] that the adhesion G of G_1 and G_2 is map self-dual.

It remains to show Claim 3. Let G^* be the dual graph of G with respect to a fixed embedding and let $\varphi: V(G) \rightarrow V(G^*)$ be a graph isomorphism between G and G^* . As G is the adhesion of G_1 and G_2 there is a unique vertex v in G belonging to G_1 and G_2 , and a unique face f incident to both graph G_1 and G_2 . Since v was chosen to be u in G_1 , it is the only vertex in G that is a cutvertex and the center of a wheel of size m . Moreover, f is the only cutvertex in G^* that can be the center of a wheel of size m . Thus φ has to map v to f . The blocks incident to v are a block with degree 3 at v stemming from G_2 , B loops also stemming from G_2 , a block consisting of a wheel of size m with center v stemming from G_1 and $3m$ attached trees stemming from G_1 . Similar the vertex f in G^* is incident to a block having degree 3 at f contained in G_1^* , a set of loops stemming from the trees in G_1 (the number of loops depends on the embedding), a wheel of size m with center f contained in G_2^* and a set of bridges stemming from the loops at G_2 . Thus, φ has to map all vertices in G_1 to vertices in G_2^* and all vertices in G_2 to vertices in G_1^* . This directly shows that G_1 and G_2 form a YES-instance of MUTUAL PLANAR DUALITY, which concludes the proof. \square

3 Decomposition Trees and the SPQR-Tree

A graph is *connected* if there exists a path between any pair of vertices. A *separating k -set* is a set of k vertices whose removal disconnects the graph. Separating 1-sets and 2-sets are *cutvertices* and *separation pairs*, respectively. A connected graph is *biconnected* if it does not have a cut vertex and *triconnected* if it does not have a separation pair. The maximal biconnected components of a graph are called *blocks*.

In the following we consider *decomposition trees* of biconnected planar graphs containing the *SPQR-trees* introduced by Di Battista and Tamassia [4, 5] as a special case. Let G be a planar biconnected graph and let $\{s, t\}$ be a *split pair*, that is either a separation pair or a pair of adjacent vertices. Let further H_1 and H_2 be two subgraphs of G such that $H_1 \cup H_2 = G$ and $H_1 \cap H_2 = \{s, t\}$. Consider the tree \mathcal{T} consisting of two nodes μ_1 and μ_2 associated with the graphs $H_1 + (s, t)$ and $H_2 + (s, t)$, respectively. For each node μ_i , the graph $H_i + (s, t)$ associated with it is the *skeleton* of μ_i , denoted by $\text{skel}(\mu_i)$, and the special directed edge (s, t) is called *virtual edge*. The edge connecting the nodes μ_1 and μ_2 in \mathcal{T} associates the virtual edge $\varepsilon_1 = (s, t)$ in $\text{skel}(\mu_1)$ with the virtual edge $\varepsilon_2 = (s, t)$ in $\text{skel}(\mu_2)$; we say that ε_1 is the *twin* of ε_2 and vice versa. Moreover, we say that ε_1 in $\text{skel}(\mu_1)$ *corresponds* to the neighbor μ_2 of μ_1 . This can be expressed as a bijective map $\text{corr}_\mu: E(\text{skel}(\mu)) \rightarrow N(\mu)$ for each node μ , where $E(\text{skel}(\mu))$ and $N(\mu)$ denote the set of edges in $\text{skel}(\mu)$ and the set of neighbors of μ in \mathcal{T} , respectively. In the example above we have $\text{corr}(\varepsilon_1) = \mu_2$ and $\text{corr}(\varepsilon_2) = \mu_1$ (the index at corr is omitted as it is clear from the context).

The above described procedure is called *decomposition* and can of course be applied further to the skeletons of the nodes of \mathcal{T} , leading to a larger tree with smaller skeletons. The decomposition can be undone by *contracting* an edge in \mathcal{T} . Let $\{\mu, \mu'\}$ be an edge in \mathcal{T} and let ε be the virtual edge in $\text{skel}(\mu)$ with $\text{corr}(\varepsilon) = \mu'$ having ε' in $\text{skel}(\mu')$ as twin. The contraction of $\{\mu, \mu'\}$ replaces these two nodes by a single node with a skeleton obtained as follows. The skeletons $\text{skel}(\mu)$ and $\text{skel}(\mu')$ are *glued* together at the twins ε and ε' according to their orientation, that is the source and target of ε is identified with the source and target of ε' , respectively. Afterwards the resulting virtual edge is removed. Applying the contraction iteratively in an arbitrary order to edges in a decomposition tree \mathcal{T} yields a tree consisting of a single node μ (which can be seen as trivial decomposition tree). Then the graph *represented* by \mathcal{T} is $\text{skel}(\mu)$, which is uniquely determined by \mathcal{T} .

A *reversed decomposition tree* is defined as a decomposition tree with the only difference that in the decomposition step one of the two twin edges is reversed and in the contraction step they are glued together such that they point in opposite directions. Note that a reversed decomposition tree can be easily transformed into an equivalent normal decomposition tree representing the same graph by reversing one virtual edge for

every pair of twin edges.

A special decomposition tree of a biconnected planar graph G is the SPQR-tree. A decomposition tree is an SPQR-tree if each inner node is either an S-, a P-, or an R-node whose skeletons contain only virtual edges forming a cycle, a bunch of at least three parallel edges or a triconnected planar graph, respectively, such that no two S-nodes and no two P-nodes are adjacent. Moreover, each leaf is a Q-node whose skeleton consists of two vertices with one virtual and one normal edge between them. The *reversed SPQR-tree* is defined analogous as a special case of the reversed decomposition tree. The SPQR-tree \mathcal{T} of a planar biconnected graph G is unique up to the reversal of pairs of virtual edges that are twins. We can assume without loss of generality that the virtual edges in the skeleton of each P-node are oriented in the same direction and those in the skeleton of each S-node form a directed cycle.

The SPQR-tree \mathcal{T} of G represents all planar embeddings of G , as there is a bijection between these embeddings and the set of all combinations of embeddings of the skeletons in \mathcal{T} . Note that the embedding choices for the skeletons consist of reordering the parallel edges in a P-node and flipping the skeleton of an R-node. The SPQR-tree of a biconnected planar graph can be computed in linear time [8]. Fixing the embeddings of skeletons in an arbitrary decomposition tree \mathcal{T} also determines a planar embedding of the represented graph G . However, there may be planar embeddings that are not represented by \mathcal{T} .

4 Succinct Representation of all Duals of a Biconnected Graph

Let G be a biconnected graph with SPQR-tree \mathcal{T} and planar embedding \mathcal{G} . In the following we study the effects of changing the embedding of G on the corresponding dual graph G^* of G . To this end, we do not consider the graphs themselves but their SPQR-trees. More precisely, we first show how the SPQR-tree of the dual graph G^* can be directly obtained from the SPQR-tree of the primal graph G . This can then be used to understand the effects in the dual graph caused by changing the embedding of a skeleton in \mathcal{T} .

We first define the *dual decomposition tree* \mathcal{T}^* of the decomposition tree \mathcal{T} representing G (with respect to a fixed embedding \mathcal{G} of G that can be represented by \mathcal{T}). It can then be shown that the dual decomposition tree represents the dual graph G^* of G . Essentially, \mathcal{T}^* is obtained from \mathcal{T} by replacing each skeleton with its directed dual and interpreting the resulting tree as a reversed decomposition tree. More precisely, for each node μ in \mathcal{T} , the dual decomposition tree \mathcal{T}^* contains a *dual node* μ^* having the dual of $\text{skel}(\mu)$ as skeleton. An edge ε^* in $\text{skel}(\mu^*)$ dual to a virtual edge ε in $\text{skel}(\mu)$ is again virtual and oriented from right to left with respect to the orientation of ε . Similarly, an edge dual to a normal edge is also a normal edge in the dual skeleton. Two virtual edges in \mathcal{T}^* are twins if and only if their corresponding primal edges are twins. This implicitly has the effect that $\text{corr}(\varepsilon)^* = \text{corr}(\varepsilon^*)$ holds. Obviously, the tree structures of \mathcal{T}^* and \mathcal{T} are isomorphic with respect to the map assigning each node in \mathcal{T} to its dual node in \mathcal{T}^* . For the case in which \mathcal{T} is the SPQR-tree of G we obtain the following. The dual of a triconnected skeleton is triconnected, the dual of a (directed) circle is a bunch of parallel edges (all directed in the same direction), and the dual of a normal edge with a parallel virtual edge is a normal edge with a parallel virtual edge. Thus, if a node μ in \mathcal{T} is an S-, P-, Q-, or R-node, its dual node μ^* in \mathcal{T}^* is a P-, S-, Q-, or R-node, respectively. This in particular shows that the dual SPQR-tree is again an SPQR-tree and not just an arbitrary decomposition tree.

Lemma 1. *Let G be a biconnected planar graph with SPQR-tree \mathcal{T} and planar embedding \mathcal{G} . The dual SPQR-tree \mathcal{T}^* with respect to \mathcal{G} is the reversed SPQR-tree of the dual G^* .*

Proof. We show a slightly more general result by dealing with arbitrary decomposition trees instead of SPQR-trees. We show that first contracting an edge $\{\mu, \mu'\}$ in a decomposition tree \mathcal{T} into a node $\mu\mu'$ and then constructing the dual decomposition tree is equivalent to first constructing the decomposition tree \mathcal{T}^* and then contracting the edge $\{\mu^*, \mu'^*\}$ into $\mu^*\mu'^*$ (recall that \mathcal{T}^* is interpreted as reversed decomposition tree, thus the gluing operation contained in the contraction of $\{\mu^*, \mu'^*\}$ is reversed). Applying this

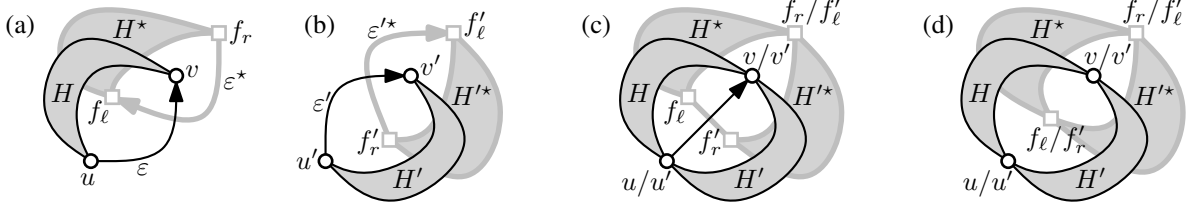


Figure 2: Glueing the virtual edge ε (a) and ε' (b) together (c) and removing the resulting edge (d).

operation iteratively until the trees \mathcal{T} and \mathcal{T}^* consist of single nodes then directly shows that the reversed decomposition tree \mathcal{T}^* represents the graph G^* dual to the graph G represented by \mathcal{T} .

Let ε and ε' be the virtual edges in $\text{skel}(\mu)$ and $\text{skel}(\mu')$ corresponding to the edge $\{\mu, \mu'\}$ in \mathcal{T} . Let further f_ℓ and f_r , and f'_ℓ and f'_r be the faces left and right to ε and ε' with respect to the orientation of ε and ε' , respectively; see Figure 2(a) and (b). We denote the graph $\text{skel}(\mu) - \varepsilon$ by H and the graph $\text{skel}(\mu^*) - \varepsilon^*$ by H^* (note that H^* is not really the dual graph of H). The graphs H' and H'^* are defined similar. When contracting $\{\mu, \mu'\}$, first the virtual edges ε and ε' are glued together, that is u and v are identified with u' and v' , respectively; see Figure 2(c). Obviously, the dual of the resulting graph can be obtained from H^* and H'^* by identifying f_r with f'_ℓ and adding the edge $\{f_\ell, f'_r\}$ (or the other way round). Finally, removing the edge (u, v) contracts f_ℓ and f'_r into a single vertex, see Figure 2(d). Thus the dual graph $\text{skel}(\mu\mu')^*$ of the resulting skeleton $\text{skel}(\mu\mu')$ can be obtained from $\text{skel}(\mu)^*$ and $\text{skel}(\mu')^*$ by removing the virtual edges ε^* and ε'^* and identifying their endpoints with each other, reversing their orientation. As this is equal to contracting $\{\mu^*, \mu'^*\}$ in \mathcal{T}^* , we have $\text{skel}(\mu\mu')^* = \text{skel}(\mu^*\mu'^*)$, which concludes the proof. \square

The above results help investigating the effects a change in the embedding of the graph G has on its dual G^* . Flipping the skeleton of an R-node and reordering the virtual edges in a P-node give rise to the following two operations: *reversal* of R-nodes and *restacking* of S-nodes. A reversal applied on an R-node μ reverses the direction of all the virtual edges in $\text{skel}(\mu)$. As no other skeleton is changed by this operation, this only affects how $\text{skel}(\mu)$ is glued to the skeletons of its adjacent nodes. Let μ be an S-node with virtual edges $\varepsilon_1, \dots, \varepsilon_k$. A restacking of μ picks an arbitrary ordering of $\varepsilon_1, \dots, \varepsilon_k$ and glues their end-points such that they create a directed cycle C in that order. Then, the skeleton of μ is replaced by C .

Lemma 2. *Let \mathcal{T} be the SPQR-tree of a biconnected planar graph and let \mathcal{T}^* be the SPQR-tree of its dual, with respect to a fixed planar embedding. Flipping an R-node and reordering a P-node in \mathcal{T} corresponds to reversing its dual R-node and restacking its dual S-node, respectively.*

Proof. Due to Lemma 1 we can work with the dual SPQR-tree instead of the SPQR-tree of the dual. Obviously, flipping an R-node μ in \mathcal{T} exchanges left and right in $\text{skel}(\mu)$ and thus reverses the orientation of each virtual edge in $\text{skel}(\mu^*)$, where μ^* is the node in \mathcal{T}^* dual to μ . Thus, flipping μ corresponds to a reversal of μ^* . Similarly, reordering the virtual edges in the skeleton of a P-node μ has the effect that the virtual edges in its dual S-node μ^* are restacked, yielding a different cycle. Note that this cycle is again directed since the virtual edges in μ are still all oriented in the same direction. \square

This shows that the SPQR-tree of the dual graph with respect to a fixed embedding can be used to represent the dual graphs with respect to all possible planar embeddings by allowing reversal and restacking operations. If we say that an SPQR-tree *represents a set of dual graphs*, we implicitly allow reversal and restacking. The following theorem directly follows.

Theorem 3. *The dual SPQR-tree of a biconnected planar graph G represents exactly the dual graphs of G .*

If we are only interested in the structure of the dual graph and not in its embedding induced by the primal graph, we may also allow the usual SPQR-tree operations, that is flipping R-nodes and reordering the virtual edges in P-nodes. In this case we can apply the reversal operation not only to R-nodes but also to P-nodes (as this only changes the embedding but not the graph). Moreover, reversing a Q-node does not change anything and the reversal of an S-node can be seen as a special way of restacking it. This observation can be used to show the following lemma.

Lemma 3. *Let G be a biconnected planar graph and let G^* be its dual graph with SPQR-tree \mathcal{T}^* with respect to an embedding \mathcal{G} of G . Let $\mathcal{T}_\varepsilon^*$ be the SPQR-tree obtained from \mathcal{T}^* by reversing the orientation of the virtual edge ε in \mathcal{T}^* and let G_ε^* be the graph represented by it. Then there exists an embedding \mathcal{G}_ε of G such that G_ε^* is the dual graph of G with respect to \mathcal{G}_ε .*

Proof. Let μ be the node in \mathcal{T}^* containing the virtual edge ε and let $\text{corr}(\varepsilon) = \mu'$ be the neighbor of μ corresponding to ε . Removing the edge $\{\mu, \mu'\}$ splits \mathcal{T}^* into two subtrees \mathcal{T}_μ^* and $\mathcal{T}_{\mu'}^*$. We claim that the reversal of all nodes in one of these subtrees (no matter which one) yields an SPQR-tree $\mathcal{T}_{\mu\mu'}^*$ representing G_ε^* . Then it follows by Lemma 2 and the observation above, that G_ε^* is a dual graph of G .

It remains to show the claim. As it does not matter whether the orientation of ε or of its twin in μ' is changed, we can assume without loss of generality that all nodes in \mathcal{T}_μ^* are reversed in $\mathcal{T}_{\mu\mu'}^*$. The graph represented by $\mathcal{T}_{\mu\mu'}^*$ can be obtained by contracting the edges in an arbitrary order. Contracting edges in the subtree $\mathcal{T}_{\mu'}^*$ has the same effect as in the original graph, since $\mathcal{T}_{\mu'}^*$ was not changed. Similarly, contracting an edge in \mathcal{T}_μ^* also has the same effect as the orientation of both corresponding virtual edges is reversed. Finally, when contracting the edge $\{\mu, \mu'\}$ the skeletons are glued together oppositely as ε is reversed whereas its twin remains the same. Thus, reversing all nodes in \mathcal{T}_μ^* is equivalent to reversing the orientation of ε , which concludes the proof. \square

Lemma 2 and Lemma 3 together yield the following theorem.

Theorem 4. *Two SPQR-trees represent the same set of dual graphs if and only if they can be transformed into each other by either using reversal and restacking operations, or by choosing an orientation of the virtual edges and restacking the skeletons of S-nodes.*

5 Equivalence Relation

We define the relation \sim on the set of planar graphs as follows. Two graphs G_1 and G_2 are related, i.e., $G_1 \sim G_2$, if and only if G_1 and G_2 can be embedded such that they have the same dual graph $G_1^* = G_2^*$. We call \sim the *common dual relation*.

Theorem 5. *The common dual relation \sim is an equivalence relation on the set of biconnected planar graphs. For a biconnected planar graph G , the set of dual graphs of G is an equivalence class with respect to \sim .*

Proof. Clearly, \sim is symmetric and reflexive. For the transitivity let G_1, G_2 and G_3 be three biconnected planar graphs such that $G_1 \sim G_2$ and $G_2 \sim G_3$. Let further $\mathcal{T}_1^*, \mathcal{T}_2^*$ and \mathcal{T}_3^* be the dual SPQR-trees representing all duals of G_1, G_2 and G_3 , respectively. Due to $G_1 \sim G_2$ there exists a graph G that is represented by \mathcal{T}_1^* and \mathcal{T}_2^* . Since the SPQR-tree of a biconnected planar graph is unique (up to the reversal of virtual edges), it follows that \mathcal{T}_1^* and \mathcal{T}_2^* are the same SPQR-trees representing the same sets of duals. The same argument shows that G_2 and G_3 have the same set of dual graphs, due to $G_2 \sim G_3$. Thus, also G_1 and G_3 have exactly the same set of dual graphs, which yields $G_1 \sim G_3$.

For the second statement, let C^* be the set of dual graphs of G . Clearly, for $G_1^*, G_2^* \in C^*$ the graph G is a common dual, thus $G_1^* \sim G_2^*$. On the other hand, let $G_1^* \in C^*$ and $G_1^* \sim G_2^*$. By the above argument, G_1^* and G_2^* have the same set of dual graphs. Thus G is a dual graph of G_2^* yielding $G_2^* \in C^*$. \square

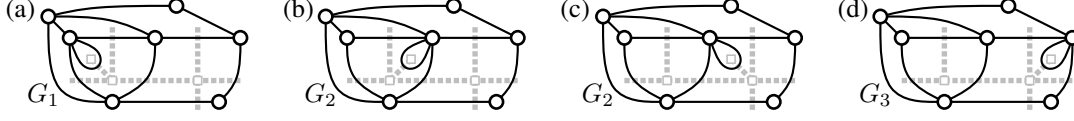


Figure 3: Illustration of Theorem 6

Theorem 5 shows that the equivalence class C of a biconnected planar graph G with respect to the common dual relation is exactly the set of dual graphs that is represented by the SPQR-tree \mathcal{T} of G . The dual SPQR-tree \mathcal{T}^* of G also represents a set of dual graphs forming the equivalence class C^* . We say that C^* is the *dual equivalence class* of C . Given an arbitrary graph $G \in C$ and an arbitrary graph $G^* \in C^*$ then G and G^* can be embedded such that they are dual to each other since C^* contains exactly the graphs that are dual to G . The problems MUTUAL PLANAR DUALITY and GRAPH SELF-DUALITY can be reformulated in terms of the equivalence classes of the common dual relation. Two biconnected planar graphs are a YES-instance of MUTUAL PLANAR DUALITY if and only if their equivalence classes are dual to each other. A biconnected planar graph is graph self-dual if and only if its equivalence class is dual to itself. This in particular means that either each or no graph in an equivalence class is graph self-dual.

Although it might seem quite natural that the common dual relation is an equivalence relation, this is not true for general planar graphs. This fact is stated in the following theorem.

Theorem 6. *The common dual relation \sim is not transitive on the set of planar graphs.*

Proof. Consider the graph G_1 consisting of a triconnected planar graph with an additional loop as depicted in Figure 3(a). Its dual graph is a triconnected component with a bridge attached to it. In the graph G_2 the loop is attached to a different vertex, see Figure 3(b). However, in both graphs G_1 and G_2 , the loop can be embedded into the same face of the triconnected component, yielding the same dual graph (with a different embedding). Thus, G_1 and G_2 have a common dual, i.e., $G_1 \sim G_2$ holds. The same argument yields that G_2 (with respect to the embedding in Figure 3(c)) and G_3 have a common dual graph, i.e., $G_2 \sim G_3$. However, G_1 and G_3 do not have a common dual for the following reason. Let v_1 and v_3 be the vertices in G_1 and G_3 incident to the loop, respectively. The only embedding choice in G_1 and G_3 is to embed the loop into one of the faces incident to v_1 and v_3 , respectively. In the dual graphs this has the effect that the bridge is attached to the corresponding vertex. Since all faces incident to v_1 have degree 3 and all faces incident to v_3 have degree 4 or 5, the resulting dual graphs cannot be isomorphic. Thus, $G_1 \not\sim G_3$ even though $G_1 \sim G_2 \sim G_3$ holds. \square

6 Solving MUTUAL PLANAR DUALITY for Biconnected Graphs

Due to Theorem 3 the problem MUTUAL PLANAR DUALITY can be rephrased as follows.

Corollary 1. *Let G_1 and G_2 be two biconnected planar graphs with SPQR-trees \mathcal{T}_1 and \mathcal{T}_2 , respectively. There is an embedding \mathcal{G}_1 of G_1 such that G_2 is dual to G_1 with respect to \mathcal{G}_1 if and only if \mathcal{T}_2 and the dual SPQR-tree \mathcal{T}_1^* represent the same set of dual graphs.*

In the following we define what it means for two SPQR-trees to be dual isomorphic and show that they are isomorphic in that sense if and only if they represent the same set of dual graphs. Afterwards, we show that testing the existence of such an isomorphism reduces to testing graph isomorphism for planar graphs, which then solves MUTUAL PLANAR DUALITY. Figure 4(a) sketches this strategy.

For two graphs G and G' with vertices $V(G)$ and $V(G')$ and edges $E(G)$ and $E(G')$, respectively, a map $\varphi: V(G) \rightarrow V(G')$ is a *graph isomorphism* if it is bijective and $\{u, v\} \in E(G)$ if and only if

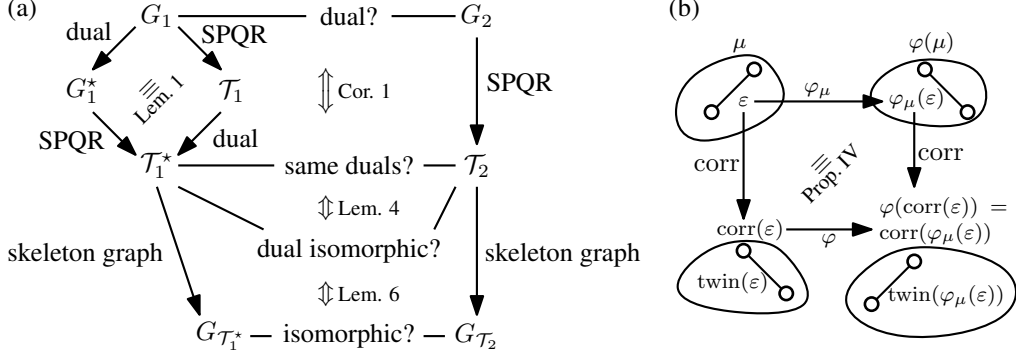


Figure 4: (a) Overview of our strategy. (b) Commutative diagram illustrating Property IV.

$\{\varphi(u), \varphi(v)\} \in E(G')$. A graph isomorphism φ induces a bijection between $E(G)$ and $E(G')$ and we use $\varphi(e)$ for $e \in E(G)$ to express this bijection. Note that we consider the edges to be undirected, thus fixing $\varphi(\cdot)$ only for the edges does not determine a map for the vertices. As the SPQR-tree has more structure than a normal tree, we require some additional properties. A *dual SPQR-tree isomorphism* between two SPQR-trees \mathcal{T} and \mathcal{T}' consists of several maps. First, a map $\varphi: V(\mathcal{T}) \rightarrow V(\mathcal{T}')$ such that

- (I) φ is a graph isomorphism between \mathcal{T} and \mathcal{T}' ; and
- (II) for each node $\mu \in V(\mathcal{T})$, the node $\varphi(\mu) \in V(\mathcal{T}')$ is of the same type.

Second, a map $\varphi_\mu: V(\text{skel}(\mu)) \rightarrow V(\text{skel}(\varphi(\mu)))$ for every R-node μ in \mathcal{T} such that

- (III) φ_μ is a graph isomorphism between $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$; and
- (IV) $\text{corr}(\varphi_\mu(\varepsilon)) = \varphi(\text{corr}(\varepsilon))$ holds for every virtual edge ε in $\text{skel}(\mu)$.

If there is a dual SPQR-tree isomorphism between \mathcal{T} and \mathcal{T}' , then we say that \mathcal{T} and \mathcal{T}' are *dual isomorphic*. Note that Property IV is a quite natural requirement and one would usually require it also for S-nodes (for P-nodes it does not make sense since every permutation is an isomorphism on its skeleton); see Figure 4(b) for a commutative diagram illustrating Property IV. However, not requiring it for S-nodes has the effect that restacking their skeletons is implicitly allowed. As the graph isomorphisms $\varphi_\mu(\cdot)$ do not care about the orientation of virtual edges it is also implicitly allowed to reverse them. These observations lead to the following lemma showing that this definition of dual SPQR-tree isomorphism is well suited for our purpose.

Lemma 4. *Two SPQR-trees represent the same set of dual graphs if and only if they are dual isomorphic.*

Proof. Let \mathcal{T} and \mathcal{T}' be two SPQR-trees representing the same set of dual graphs. By Theorem 4 this implies that they can be transformed into each other using reversal and restacking operations. Clearly, the identity map, mapping \mathcal{T} and each of its skeletons to itself, is a dual SPQR-tree isomorphism. It remains a dual SPQR-tree isomorphism when restacking an S-node, since Properties I, II are independent from the skeletons and Properties III, IV are only required for R-nodes. Moreover, the reversal of an R-node preserves Properties I–IV since our definition of graph isomorphism considers edges to be undirected. It follows that \mathcal{T} and \mathcal{T}' are dual isomorphic.

For the opposite direction assume that φ together with $\varphi_{\mu_1}, \dots, \varphi_{\mu_k}$ is a dual SPQR-tree isomorphism from \mathcal{T} to \mathcal{T}' . For every virtual edge ε in an R-node μ the map φ_μ determines whether the orientation of ε has to be reversed to match the orientation of $\varphi_\mu(\varepsilon)$. Moreover, how φ maps the neighbors of an S-node μ to the neighbors of $\varphi(\mu)$ determines a restacking operation transforming $\text{skel}(\mu)$ into $\text{skel}(\varphi(\mu))$. It follows that \mathcal{T} can be transformed into \mathcal{T}' by applying restacking and reversal operations. Hence, \mathcal{T} and \mathcal{T}' represent the same set of dual graphs, which concludes the proof. \square

In the following we show how the question of whether two SPQR-trees are dual isomorphic can be reduced to the graph isomorphism problem on planar graphs, which can be solved in linear time [6]. We

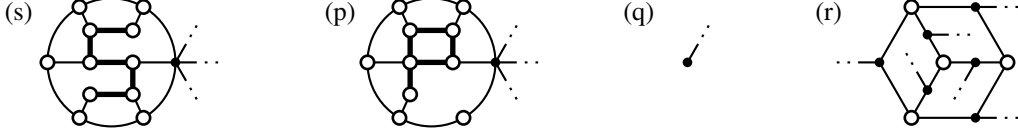


Figure 5: The subgraphs H_μ of the skeleton graph depending on the type of the node μ . The small black vertices are the attachment vertices.

define the *skeleton graph* $G_{\mathcal{T}}$ of an SPQR-tree \mathcal{T} as follows. For each node μ in \mathcal{T} there is a subgraph H_μ in $G_{\mathcal{T}}$ and for each edge $\{\mu, \mu'\}$ in \mathcal{T} the skeleton graph contains an edge connecting H_μ and $H_{\mu'}$. In the following we describe the subgraphs H_μ for the cases that μ is an S-, a P-, a Q- or an R-node and define *attachment vertices* that are incident to the edges connecting H_μ to other subgraphs.

If μ is an S- or a P-node, the subgraph H_μ contains only one attachment vertex v_μ and all subgraphs stemming from neighbors of μ are attached to v_μ . To distinguish between S- and P-nodes, small non-isomorphic subgraphs called *tags* are additionally attached to v_μ , see Figure 5(s) and (p), respectively. If μ is a Q-node, then H_μ is a single attachment vertex, see Figure 5(q). Note that μ is a leaf in \mathcal{T} and thus H_μ is also a leaf in $G_{\mathcal{T}}$. If μ is an R-node, H_μ is the skeleton $\text{skel}(\mu)$, where additionally every virtual edge ε is subdivided by an attachment vertex v_ε , see Figure 5(r) for the case in which $\text{skel}(\mu)$ is K_4 . The subgraph $H_{\text{corr}(\varepsilon)}$ stemming from the neighbor $\text{corr}(\varepsilon)$ of μ is attached to H_μ over the attachment vertex v_ε .

Lemma 5. *The skeleton graph of an SPQR-tree is planar and can be computed in linear time.*

Proof. Clearly, the skeleton graph $G_{\mathcal{T}}$ of an SPQR-tree \mathcal{T} can be computed in linear time by processing each node μ separately to compute the subgraph H_μ consuming time linear in the size of $\text{skel}(\mu)$. Note that this implicitly shows that the size of $G_{\mathcal{T}}$ is linear.

Let \mathcal{T} be an SPQR-tree rooted at an arbitrary node. The skeleton graph $G_{\mathcal{T}}$ can be embedded in a planar way by embedding the subgraphs corresponding to the nodes in \mathcal{T} top-down with respect to the chosen root. Obviously, every subgraph in $G_{\mathcal{T}}$ corresponding to a node in \mathcal{T} is planar, thus we can start by embedding the subgraph corresponding to the root arbitrarily. Let μ be a non-root node in \mathcal{T} and let μ' be its parent. If μ is not an R-node, H_μ can be embedded with its only attachment vertex on the outer face. If μ is an R-node, H_μ can be embedded with the attachment vertex corresponding to the parent μ' of μ in \mathcal{T} on the outer face. Thus, in any case, H_μ can be placed inside a face incident to the attachment vertex stemming from μ' corresponding to μ , yielding a planar drawing. \square

Lemma 6. *Two SPQR-trees are dual isomorphic if and only if their skeleton graphs are isomorphic.*

Proof. Let \mathcal{T} and \mathcal{T}' be two SPQR-trees and let φ together with $\varphi_{\mu_1}, \dots, \varphi_{\mu_k}$ be a dual SPQR-tree isomorphism between them. We show how this induces a graph isomorphism φ_G between the skeleton graphs $G_{\mathcal{T}}$ and $G_{\mathcal{T}'}$. If μ is an S-, P- or Q-node, then its corresponding subgraph in H_μ only contains a single attachment vertex v_μ . Since $\varphi(\mu)$ is of the same type (due to Property II of dual SPQR-tree isomorphisms), the subgraph $H_{\varphi(\mu)}$ also contains a single attachment vertex $v_{\varphi(\mu)}$ and we set $\varphi_G(v_\mu) = v_{\varphi(\mu)}$. For S- and P-nodes we additionally simply map their tags isomorphically to one another. For the case that μ is an R-node, the map φ_μ is a graph isomorphism between $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$ (Property III). Thus, it induces a graph isomorphism between H_μ and $H_{\varphi(\mu)}$ since these subgraphs are obtained from $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$, respectively, by subdividing each virtual edge. It remains to show that φ_G respects the edges between attachment vertices of different subgraphs. Since φ is a graph isomorphism (Property I), attachment vertices of two subgraphs of $G_{\mathcal{T}}$ are connected if and only if the corresponding subgraphs in $G_{\mathcal{T}'}$ are connected. Moreover, Property IV ensures that for a subgraph stemming from an R-node the right attachment vertices are chosen (for other nodes this is clear since their subgraphs have unique attachment vertices).

For the opposite direction, assume φ_G is a graph isomorphism between $G_{\mathcal{T}}$ and $G_{\mathcal{T}'}$. Let H_μ be the subgraph stemming from a node μ in \mathcal{T} . As H_μ is a block in $G_{\mathcal{T}}$ (or a leaf if μ is a Q-node), it has to be mapped to a block in $G_{\mathcal{T}'}$. As all edges in $G_{\mathcal{T}'}$ connecting attachment vertices of subgraphs stemming from different nodes are bridges, all vertices in H_μ have to be mapped to vertices in $H_{\mu'}$ for some node μ' in \mathcal{T}' . This defines the map φ by setting $\varphi(\mu) = \mu'$. Clearly, φ is a graph isomorphism between \mathcal{T} and \mathcal{T}' , since two subgraphs in a skeleton graph are connected by an edge if and only if the corresponding nodes in its SPQR-tree are adjacent, thus, φ satisfies Property I. Since the only leaves in a skeleton graph stem from Q-nodes, $\varphi(\mu)$ is a Q-node if and only if μ is a Q-node. Let v be an attachment vertex stemming from an inner node μ in \mathcal{T} . Then v is a cutvertex and, since every cutvertex in a skeleton graph is an attachment vertex, $\varphi_G(v)$ is also an attachment vertex in $G_{\mathcal{T}'}$. The vertex v has degree 3 if and only if μ is an R-node, thus φ maps R-nodes to R-nodes. Moreover, if μ is an S-node, v cannot be mapped to an attachment vertex stemming from a P-node, since the tags attached to S- and P-nodes are not isomorphic. Hence, φ maps S- and P-nodes to S- and P-nodes, respectively, and thus satisfies Property II.

To obtain a dual SPQR-tree isomorphism, it remains to define a map φ_μ for each R-node μ in \mathcal{T} that satisfies Properties III and IV. As observed before, φ_G defines a bijection between the vertices in the subgraph H_μ stemming from μ and the vertices in $H_{\varphi(\mu)}$ stemming from $\varphi(\mu)$. As H_μ and $H_{\varphi(\mu)}$ are the skeletons $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$ (with a subdivision vertex on each virtual edge), φ_G defines a graph isomorphism φ_μ between $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$ (satisfying Property III). To show that Property IV holds consider a virtual edge ε in $\text{skel}(\mu)$ and the attachment vertex v_ε in H_μ stemming from it. Let further denote $v_{\text{corr}(\varepsilon)}$ the attachment vertex in $H_{\text{corr}(\varepsilon)}$ such that $G_{\mathcal{T}}$ contains the edge $\{v_\varepsilon, v_{\text{corr}(\varepsilon)}\}$. Then φ_G maps $\{v_\varepsilon, v_{\text{corr}(\varepsilon)}\}$ to an edge $\{\varphi_G(v_\varepsilon), \varphi_G(v_{\text{corr}(\varepsilon)})\}$ in $G_{\mathcal{T}'}$. Since $\varphi_G(v_\varepsilon) = v_{\varphi_\mu(\varepsilon)}$ holds by the definition of φ_μ , and $\varphi_G(v_{\text{corr}(\varepsilon)})$ stems from the node $\varphi(\text{corr}(\varepsilon))$ by the definition of φ , we have that $\text{corr}(\varphi_\mu(\varepsilon)) = \varphi(\text{corr}(\varepsilon))$ by the definition of the skeleton graph $G_{\mathcal{T}'}$. As this establishes Property IV, it concludes the proof. \square

Theorem 7. *MUTUAL PLANAR DUALITY can be solved in linear time for biconnected planar graphs.*

Proof. See Figure 6 for an example. Corollary 1 states that MUTUAL PLANAR DUALITY can be solved by testing whether two SPQR-trees (that can be computed in time linear in the size of the input graphs [8]) represent the same set of dual graphs. By Lemma 4 it is equivalent to test whether these two SPQR-trees are dual isomorphic, which can be done by testing whether their skeleton graphs are isomorphic, due to Lemma 6. The skeleton graph of an SPQR-tree is planar and has linear size, see Lemma 5. Hence, we can use the linear time algorithm for testing whether two planar graphs are isomorphic by Hopcroft and Wong [6] yielding a linear time algorithm solving MUTUAL PLANAR DUALITY. \square

Corollary 2. *GRAPH SELF-DUALITY can be solved in linear time for biconnected planar graphs.*

7 Conclusion

In this paper we defined and studied the problem MUTUAL PLANAR DUALITY of testing whether, given two graphs G_1 and G_2 , there exists an embedding of G_1 such that the corresponding dual graph is isomorphic to G_2 . We proved that MUTUAL PLANAR DUALITY is NP-complete in the general case, while it is solvable in polynomial (actually linear) time for biconnected planar graphs.

The interest on this problem is twofold. On one hand, it represents a new step in the fundamental theory of planar graphs isomorphism, also testified by the fact that, as a side effect, it provides the same results for the well-known problem of testing GRAPH SELF-DUALITY [9, 2]. On the other hand, it could be seen as a single example among a plethora of problems that require to find a dual graph of G_1 satisfying certain properties. In this direction, we believe that the definition of the new data-structure *dual SPQR-tree* and of the operations that can be applied on it to efficiently handle all the duals of a biconnected planar graph could

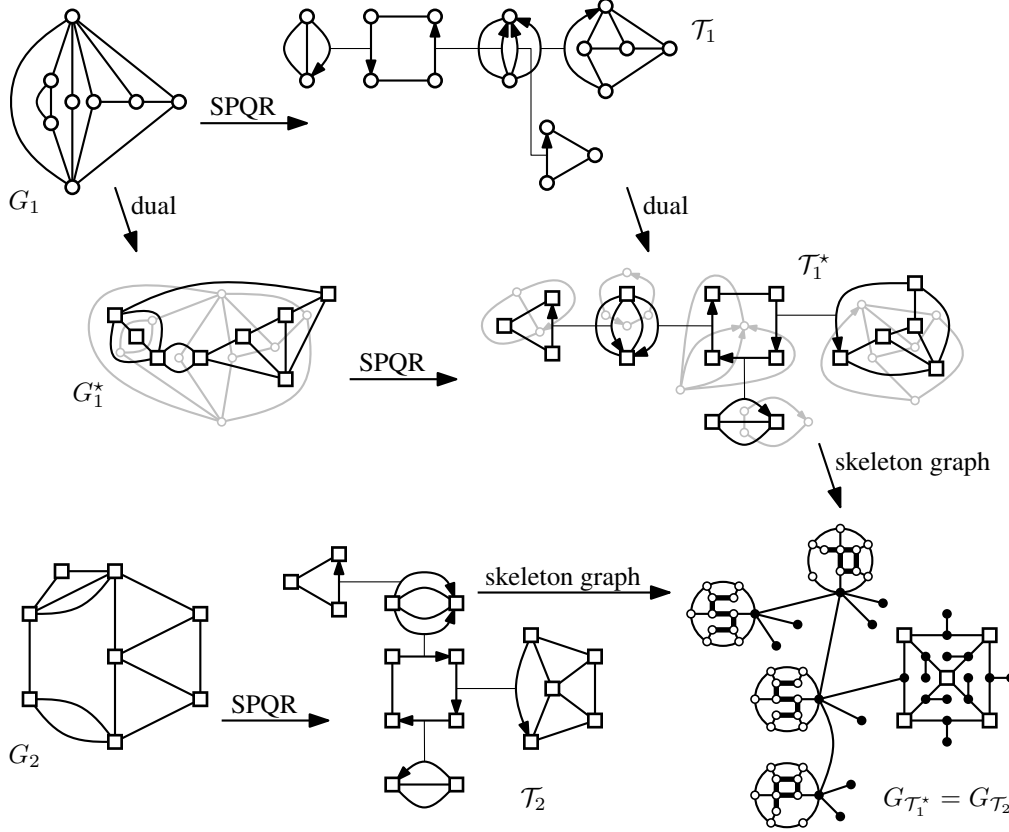


Figure 6: First building the dual graph G_1^* of G_1 (with respect to a fixed embedding) and then building its SPQR-tree or first building its SPQR-tree \mathcal{T}_1 and then its dual SPQR-tree yields the same tree \mathcal{T}_1^* (Lemma 1). The graphs G_1 and G_2 are dual to each other (with respect to at least one pair of embeddings) if and only if \mathcal{T}_1^* and \mathcal{T}_2 represent the same set of duals (Corollary 1), which is the case if and only if their skeleton graphs $G_{\mathcal{T}_1^*}$ and $G_{\mathcal{T}_2}$ are isomorphic (Lemma 4 and Lemma 6).

be considered as a main result of this paper, independently of its application to solve MUTUAL PLANAR DUALITY, since it could potentially be used to tackle many other problems of the same type.

As remarked above, the results we obtained on MUTUAL PLANAR DUALITY can be extended to GRAPH SELF-DUALITY, asking whether a given graph G can be embedded in such a way that the corresponding dual is isomorphic to G . The restricted version MAP SELF-DUALITY [10] of GRAPH SELF-DUALITY requires the embedding of G to be preserved in the isomorphism with the corresponding dual. We could prove that the NP-completeness result for MUTUAL PLANAR DUALITY extends to MAP SELF-DUALITY, but we could not prove the same for the polynomial-time testing algorithm. Hence, we leave as an open problem the question whether MAP SELF-DUALITY can be solved efficiently for biconnected planar graphs.

References

- [1] P. Angelini, G. Di Battista, and M. Patrignani. Finding a minimum-depth embedding of a planar graph in $O(n)$ time. *Algorithmica*, 60:890–937, 2011.
- [2] D. Archdeacon and R. B. Richter. The construction and classification of self-dual spherical polyhedra. *Journal of Combinatorial Theory, Series B*, 54(1):37–63, 1992.
- [3] D. Bienstock and C. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. *Algorithmica*, 5:93–109, 1990.
- [4] G. Di Battista and R. Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15(4):302–318, 1996.
- [5] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25:956–997, 1996.
- [6] J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the sixth annual ACM symposium on Theory of computing*, STOC '74, pages 172–184. ACM, 1974.
- [7] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [8] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In J. Marks, editor, *Graph Drawing (GD '00)*, volume 1984 of *LNCS*, pages 77–90, 2001.
- [9] B. Servatius and P. R. Christopher. Construction of self-dual graphs. *Am. Math. Monthly*, 99(2):153–158, 1992.
- [10] B. Servatius and H. Servatius. Self-dual graphs. *Discrete Math.*, 149(1-3):223–232, 1996.
- [11] W. T. Tutte. Connectivity in matroids. *Canad. J. Math.*, 18:1301–1324, 1966.
- [12] H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1):150–168, 1932.